UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,331	12/05/2003	Teik-Chung Tan	5500-91700	1155

53806 7590 08/22/2007
MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL (AMD)
P.O. BOX 398
AUSTIN, TX 78767-0398

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
08/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/729,331
Filing Date: December 05, 2003
Appellant(s): TAN ET AL.

Robert C. Kowert, Reg. #39,255
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/5/2007 appealing from the Office action mailed 10/3/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

4,338,661	Tredennick et al.	7-1982
5,761,470	Yoshida	6-1998
6,023,757	Nishimoto et al.	2-2000

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-3, 5-10, 12-14, 16-21, 23-24, and 27 are rejected under 35 U.S.C. 102(b) as being anticipated by Tredennick et al. (USPN 4,338,661, herein Tredennick).

3. As per Claim 1, Tredennick teaches: A microprocessor, comprising:

a microcode ROM (Column 15, Lines 33-34), wherein a row in the microcode ROM stores a plurality of groups of microcode operations (Column 15, Lines 36-37), wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine (Column 1, Lines 55-56), and wherein the row stores an associated

control sequence for each of the plurality of groups of microcode operations (Column 15, Lines 52-55 and Lines 59-66); and

a control sequence logic unit coupled to the microcode ROM (Figure 4, Address Selection 64), wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control, sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify an other row storing one or more next groups of microcode operations comprised in the microcode routine(Column 15, Lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in Column 15 Lines 52-55 and 59-66).

4. As per Claim 2, Tredennick teaches: The microprocessor of claim 1, wherein at least one of the plurality of groups of microcode operations stored in the row is part of a different microcode routine (Figure 11, also see Column 19, Lines 52-55).

5. As per Claim 3, Tredennick teaches: The microprocessor of claim wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine based on information contained in the control sequence associated with the group of microcode operations stored in the row (Column 15, Lines 52-55 and 59-66, which defines which row and position the next group is located).

6. As per Claim 5, Tredennick teaches: The microprocessor of claim 1, wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations (Column 15, Lines 49-55 teach a microcode instruction for branches, and Column 17 Lines 29-32 show that the outcome either way will be in the same row specified by the control sequence).

7. As per Claim 6, Tredennick teaches: The microprocessor of claim 1, wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments (Column 19, Lines 22-27. It is disclosed that for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a different number of groups than the other lines).

8. As per Claim 7, Tredennick teaches: The microprocessor of claim 6, wherein groups of microcode operations stored in a same one of the plurality of segments have a same maximum width (Column 19, Lines 22-27).

9. As per Claim 8, Tredennick teaches: The microprocessor of claim 7, wherein groups of microcode operations stored in one of the plurality of segments have a maximum width that is different from a maximum width of groups of microcode operations stored in another one of the plurality of segments (Column 19, Lines 22-27).

10. As per Claim 9, Tredennick teaches: The microprocessor of claim 8, wherein one of the plurality of segments stores one group of microcode operations and one associated control sequence per row (Column 19, Lines 22-27).

11. As per Claim 10, Tredennick teaches: The microprocessor of claim 6, wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations and a position of one or more control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations (Column 15, Lines 52-55 and 59-66, which defines which row and position the next group is located).

12. As per Claim 12, Tredennick teaches: A computer system, comprising:

a system memory (Column 4, Lines 54-56); and

a microprocessor coupled to the system memory (Column 4, Lines 51-52), comprising;

a microcode ROM (Column 15, Lines 33-34), wherein a row in the microcode ROM stores a plurality of groups of microcode operations (Column 15, Lines 36-37),

wherein one of the plurality of groups of microcode operations is comprised in a particular microcode routine (Column 1, Lines 55-56), and wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations (Column 15, Lines 52-55 and Lines 59-66); and

a control sequence logic unit coupled to the microcode ROM (Figure 4, Address Selection 64), wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify an other row storing one or more next groups of microcode operations comprised in the microcode routine (Column 15, Lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in Column 15 Lines 52-55 and 59-66).

13. As per Claim 13, Tredennick teaches: The computer system of claim 12, wherein at least one of the plurality of groups of microcode operations stored in the row is part of a different microcode routine (Figure 11, also see Column 19, Lines 52-55).

14. As per Claim 14, Tredennick teaches: The computer system of claim 12, wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, based on information contained in the control sequence associated with the group of microcode operations stored in the row (Column 15, Lines 52-55 and 59-66, which defines which row and position the next group is located).

15. As per Claim 16, Tredennick teaches: The computer system of claim 12, wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on branch prediction as well as the control sequence associated with the group of microcode operations (Column 15, Lines 49-55 teach a microcode instruction for branches, and Column 17 Lines 29-32 show that the outcome either way will be in the same row specified by the control sequence).

16. As per Claim 17, Tredennick teaches: The computer system of claim 12, wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of any of the plurality of microcode ROM segments, and wherein the number of groups of microcode operations stored in a row in one of the plurality of microcode ROM segments differs from the number of groups of microcode operations stored in a row in another one of the plurality of microcode ROM segments (Column 19, Lines 22-27. It is disclosed that for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a different number of groups than the other lines).

17. As per Claim 18, Tredennick teaches: The computer system of claim 17, wherein groups of microcode operations stored in any one of the plurality of microcode ROM segments have a same maximum width (Column 19, Lines 22-27).

18. As per Claim 19, Tredennick teaches: The computer system of claim 18, wherein groups of microcode operations stored in one of the plurality of microcode ROM segments have a maximum width that is different from a maximum width of groups of microcode operations stored in another one of the plurality of microcode ROM segments (Column 19, Lines 22-27).

19. As per Claim 20, Tredennick teaches: The computer system of claim 19, wherein one of the plurality of microcode ROM segments stores one group of microcode operations and one associated control sequence per row (Column 19, Lines 22-27).

20. As per Claim 21, Tredennick teaches: The computer system of claim 17, wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations within a row and their associated control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations (Column 15, Lines 52-55 and 59-66, which defines which row and position the next group is located).

21. As per Claim 23, Tredennick teaches: A method, comprising:

storing a plurality of groups of microcode operations (Column 15, Lines 36-37) and a plurality of control sequences in a row in a microcode ROM, wherein each of the plurality of control sequences is associated with a respective one of the groups of microcode operations (Column 15, Lines 52-55 and Lines 59-66); and

in response to accessing one of the plurality of groups of microcode operations, using the one of the plurality of control sequences associated with that one of the plurality of groups to identify a next group of microcode operations to output from the microcode ROM (Column 15, Lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in Column 15 Lines 52-55 and 59-66).

22. As per Claim 24, Tredennick teaches: The method of claim 23, further comprising identifying the next group of microcode operations based on one or more branch predictions as well as the one of the plurality of control sequences if the one of the plurality of groups of microcode operations includes one or more branch operation (Column 15, Lines 49-55 teach a microcode instruction for branches, and Column 17 Lines 29-32 show that the outcome either way will be in the same row specified by the control sequence).

23. As per Claim 27, Tredennick teaches: A system, comprising:

a microcode ROM (Column 15, Lines 33-34), wherein a row in the microcode ROM stores a plurality of groups of microcode operations (Column 15, Lines 36-37) and

wherein the row stores an associated control sequence for each of the plurality of groups (Column 15, Lines 52-55 and Lines 59-66); and

means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM (Column 15, Lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in Column 15 Lines 52-55 and 59-66).

Claim Rejections - 35 USC § 103

24. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

25. Claims 4, 11, 15, 22, and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tredennick, in view of Yoshida (USPN 5,761,470).

26. As per Claim 4, Tredennick teaches the microprocessor of claim 3, but fails to teach: wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Yoshida also teaches that the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (Column 1, Lines 51-56). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

27. As per Claim 11, Tredennick teaches the microprocessor of claim 1, but fails to teach: wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time

the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

28. As per Claim 15, Tredennick teaches the computer system of claim 14, but fails to teach: wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Yoshida also teaches that the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (Column 1, Lines 51-56). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

29. As per Claim 22, Tredennick teaches the computer system of claim 12, but fails to teach: wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM and comprised in the microcode routine are output during a single access.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

30. As per Claim 25, Tredennick teaches the method of claim 23, but fails to teach: further comprising substituting NOPs for one or more groups of microcode instructions stored in a same row as the next group of microcode operations dependent on the one of the plurality of control sequences.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Yoshida also

teaches that the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (Column 1, Lines 51-56). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

31. As per Claim 26, Tredennick teaches the method of claim 25, but fails to teach: wherein the groups of microcode operations comprised in the microcode routine and the NOPs are output as a single line.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously, where one VLIW word can specify a plurality of instructions (Column 1, Lines 25-30), much as a microcode ROM row specifies a plurality of instructions (groups). This parallelism allows the processor to execute at a high speed, allowing for faster performance (Column 1, Lines 17-20). Given the advantage of higher speed through parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance.

(10) Response to Argument

32. Regarding Claims 1, 2, 12, 13, and 23, Appellant has argued that a single instruction is not a "group of microcode operations". Now, when Examiner wrote that

particular statement, he had interpreted the statement of "a plurality of groups of microcode operations", as such that there are multiple groups, and multiple instructions, and not necessarily multiple instructions in each group. However, Examiner concedes that in the second limitation, it does state that a group has operations (plural). However, this is a moot argument, because Examiner notes that the Appellant has not only taken Examiners statement out of context, but has actually completely ignored the actual rejection to the claim. What Examiner actually said, in context, is that there are groups with multiple instructions, such as the swap and tasm instructions with Appellant mentioned but did not expand upon, and that **alternatively,** the groups could contain a single instruction, with the interpretation given above. Appellant has argued the Examiners alternative case, but has completely ignored the actual rejection, and has not made any argument towards the Examiner explicitly pointing out where groups in Tredennick have multiple instructions. Therefore, even as Examiner may be mistaken in one line of the after-final response in which multiple explanations were given, because Appellant has decided not to argue the actual rejection of the claim, Examiner feels the rejection should stand. It is very clear that in Figure 11a, the row "00 0100 11" contains two groups, each containing two instructions, swap, and tasm (each two part instructions). Examiner further notes that Appellant has given no definition of what a group is, other than it simply contains multiple operations, and has a "control sequence".

Further regarding Claim 1, Appellant has argued that Tredennick does not teach "an associated control sequence for each of the plurality of groups of microcode operations". Appellant proceeds to indicate that each micro/nano word contains an

address for the next word to be accessed. By doing so, Appellant has validated Examiners rejection of the claim. The control sequence as claimed identifies an other row where one or more next groups of the routine reside, which is exactly what Appellant has stated Tredennick teaches. Examiner notes that the claim is a *comprising* claim, such that the microprocessor must contain the elements, but does not preclude other elements from being present. Tredennick simply has multiple control sequences associated with each group, which clearly fits the claim language of the group having an associated control sequence, in the comprising language.

Appellant has also made the argument that these control sequences do not identify another row storing one or more next groups comprised in the microcode routine, but instead discloses "the next address to be decoded for the control stores". As can clearly be seen from Tredennick's figures, for example, 11a, **every** address specifies a row. To access an instruction, a row and column must be specified, using the address, where the high 8 bits indicate the row, and the low 2 bits indicate the column. Therefore, every control sequence in Tredennick teaches a row where the next instruction (which is in a group, thus specifies a group) resides. Appellant has further argued on page 12 that these next address sequences teach an "other" row. However, referring to Column 17, and Appendix A (which Column 17 describes), it can be seen that there is a listing of the destinations of each of these instructions, which point to a specific row to go to, which are not in the same row as the current instruction. For example, in Column 17, Lines 29-32, a conditional branch is given as an example, with 2 possible outcomes. Both outcomes must be in the same row, however, this is not the

same row as the branch is in, as can be seen in Appendix A. Furthermore, Appellant seems to be arguing that when executing a program (microcode routine), that Tredennick somehow can only use the instructions on a single row, and not include other instructions in the microcode routine, which is clearly not the case, as seen above. When the swap or trap instructions are complete, some other instruction will be executed, which is on a different row. It appears Appellant is trying to read detail from his specification into the claims as to what a microcode routine is (and Appellant seems to be implying that a microcode routine is a single instruction, which is not the common usage of the term), when it has not been defined in any way in the claims, and has been interpreted by the Examiner in the broadest reasonable way, which is a computer program.

Additionally, Appellant has argued that for each individual operation, a next address is provided for a next individual operation, and not an other row. However, as stated before, and as is very clear from Tredennick's figures, every operation is on a row, and therefore, to address an instruction, the row must be identified.

33. Regarding Claim 27, Appellant has argued that Tredennick does not teach the "means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing an ext group of microcode operations stored in the microcode ROM". However, as explained above, if each control sequence identifies an instruction, and each instruction is in a group, then the identification of the instruction identifies the group as well. Given that group has not

been given any kind of definition, Examiner feels that by accessing an instruction in a group, then the group was clearly accessed as well. Examiner also notes for Claim 27, that there is no explicit indication that a group contains multiple instructions, when interpreted as Examiner explained in Claim 1 above, that there are "groups" of "operations", but that does not necessarily indicate that each group contains multiple operations, and thus, in this claim, a single instruction can be a group, thus Tredennick even more clearly anticipates the claim.

34. Regarding Claims 3 and 14, Appellant has made the argument, as mentioned before, that identifying the next address, or "row and position" has anything to do with identifying which of a plurality of groups stored in the other row of the ROM are comprised in the microcode routine, however, as stated above, Examiner believes that by identifying the address of the next instruction, which is in a group, that the group is identified as well, and thus fits the claim language. Appellant's arguments for Claims 5, 16, and 24 are essentially the same, by identifying the next microword, Tredennick identifies a row and group in the process.

Appellant has further argued Tredennick does not teach branch prediction, and thus cannot anticipate the claim. However, Examiner is taking the stance that because branch prediction has been such an integral and important part of modern processors, going back decades, and providing such a large and important performance increase, that Tredennick does utilize branch prediction, and it is not mentioned because it is such a basic and well-known concept that there is no reason to disclose it in a patent,

because everyone does it and everyone knows what it is. With branch resolution being one of the biggest latency problems with computers (along with memory access times), it would be foolish for one to not use branch prediction in a processor, and given that Tredennick already has outcomes for both the taken and not taken branches, Examiner believes that Tredennick makes use of branch prediction, because as stated earlier, branch prediction is one of the most basic and fundamental concepts of modern computer architecture.

35. Regarding Claims 6 and 17, Appellant has made the argument that a "word line" in Tredennick "selects a single microword and/or a single nanoword" (emphasis added by Appellant). Appellant has further argued that Examiner is incorrect in stating that a row in the micro ROM corresponds to multiple "words" in the nano ROM. Appellant has used a citation from Tredennick to prove his case, which states: "the same address is presented to the decoders of both the micro ROM and the nano ROM. For any input address, there will be no more than one world line in each ROM which remains high. The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and output columns" (again, emphasis by Appellant). However, Appellant is ignoring a crucial part of this citation, and is in fact attempting to twist the interpretation of this citation by emphasizing incorrect parts of the sentences. Now, looking at the Appellants original argument, which states that a word line can only select a single nanoword, it is abundantly clear that Tredennick states the

exact opposite. Looking again at the citation, which clearly states: "The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and output columns" (emphasis by Examiner). It is extremely clear from this citation that the output word is generated by **the intersection of the word line and the column**, where the word line is **the row specifying multiple operations**. Appellant has completely ignored the actual teachings of the section he cited, and is attempting to focus on different words of the citation to prove a statement that runs contradictory to the teachings of Tredennick. In fact, looking at the citation, there is no room for question or error that Tredennick teaches what Examiner claims, that a word line corresponds to a **row**, and the selected output word is **the intersection of the row and column**. Therefore, it is very clear that Appellant is incorrect in his argument, and Examiner is correct, and there can be no other interpretation of the cited section that says otherwise, as Tredennick is very clear on what a word line is.

36. Regarding Claims 7-9 and 18-20, Appellant has argued that Tredennick does not teach the limitation of a segment having a maximum width, or what is stored in a segment (and has further indicated that in the claimed invention, a row contains one group of microcode operations and one associated control sequence per row, which has not been claimed, and in fact, Appellant has attempted to argue the exact opposite interpretation, so Examiner is unclear why Appellant is attempting to redefine what he has claimed in this section of arguments, with absolutely no basis for the definition in

the claims). The portion of Tredennick cited by the Examiner states that "Each word line in the nano ROM however may represent one, two, or four possible different input addresses. In the preferred embodiment of the data processor, a word line in the nano ROM may represent as many as right different input addresses". Examiner interprets this section in that each word line may contain a different number of operations, and the lines that contain the same number of operations fit into what has been claimed as a "segment", as they all have the same "maximum width".

37. Regarding Claims 10 and 21, Appellants arguments have already been addressed by the preceding remarks, so Examiner will refer back to the previous remarks for brevity.

38. Regarding Claims 4, 15, and 25, Appellants have argued that Yoshida cannot be combined with Tredennick because it would change the principle of operation. Examiner disagrees, and has argued that the principle of operation is to execute instructions, and finding ways to do so more efficiently does not change that principle. Appellants have argued that the amount of changes required to combine the references would change the principle of operation. However, Examiner does not believe this is a valid argument, because according to the Appellant's argument, there is no way to reject a patent application involving any kind of enhancement to a computer system that requires a change in the way some things are handled, because it would necessarily change the principle of operation. Therefore, by Appellants logic, adding a cache, a branch

prediction unit, or a reservation station would change the principle of operation of a processor, and thus would not have been obvious, which is a fundamentally flawed argument. The very nature of computers requires some changes in order to make use of new technology, especially in regard to parallelism, however, these features are extremely well known in the art, and one of ordinary skill in the art would know of them, and know how to make use of them in a prior art machine to take advantage of them. The addition of VLIW processing to Tredennick's invention is no different then adding a caching system, yet Appellant would argue that this somehow fundamentally changes the principle of operation of the device, because the machine must be altered to make use of the caching system. Examiner strongly disagrees with this line of thinking, and believes the addition of Yoshida is proper, and that Appellant's interpretation of the principle of the invention is flawed, otherwise, it would appear that 35 U.S.C. 103 cannot be applied to any computer reference, because almost every addition to a computer changes the way the rest of the machine works. Furthermore, Examiner believes Yoshida would not require as much effort to be combined with Tredennick as other references may, because they are analogous in several ways. Namely, a VLIW word is essentially a long instruction word with multiple instructions on it, which is very similar to the nano ROM word of Tredennick's invention. The entire line could be grabbed and used as a VLIW word; Yoshida's Figure 1 shows that there is a fair amount of similarity between the two inventions.

Appellant has further argued that the combination would not result in the claimed invention, because Yoshida allegedly teaches that NOP's are inserted where no

operations are available to be executed in parallel, and not when operations from another microcode routine are output to the scheduler. However, Examiner feels that this fits the claim language, if the other operations are part of a different microcode routine, then they aren't executable in parallel with the running routine, thus would be replaced with NOP's.

Regarding Appellant's arguments regarding Claims 11, 22, and 26, the arguments presented for these claims are substantially similar to the arguments presented above, and Examiner refers to the previous remarks for the sake of brevity.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert Fennema

Conferees:

Eddie Chan

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Lynne Browne
APPEAL PRACTICE SPECIALIST, TQAS
TECHNOLOGY CENTER 2100